Alexandria University

Faculty of Engineering

Computer and Systems Engineering

First Year

CS 111: Probability

Fall 2021

Lab 2

Overview

R is best used at data science, hence data and data manipulation are the core of any R application. Data is generally represented in tables (Data Frames) where rows present different instances and columns represent different attributes. Data manipulation are operations including search, select, filter, group, transform.

Loading Data

Data can be presented in various forms (.txt,.csv,.h5, ….etc), we use read function to read data from files.

```
DataFrame = read.csv('data.csv')
```

To read a space separated file use:

```
DataFrame = read.csv('data.csv',sep=' ')
```

Loading ready data from R:

R has some ready data to be used called "built in Datasets". It can be loaded by first loading the `datasets` library using:

```
library(datasets)
```

then loading the data using `data` function, for example to load a built in data called iri write the following command: `data("iris")` , then `iris` variable can be used like `DataFrame` in above case.

Displaying Data

- Dataframes can be printed.

  ```
  print(DataFrame)
  ```

- We can use **head** to print the first rows only of a huge data frame

  ```
  print(head(DataFrame))
  ```

- To make **head** print the first ‘n’ rows, you can pass ‘n’ as a second argument to head function. For example, to print the first 5 rows of DataFrame use the following command:

  ```
  print(head(DataFrame, 5))
  ```

- We can use **names** to print the columns of data frame

  ```
  print(names(DataFrame))
  ```

- We can use **str** to print the dimensions and data types (classes) of columns of data frame

  ```
  print(str(DataFrame))
  ```

Filtering, Selection and Indexing

Three fundamental operations are data **filtering**, **selection** and **indexing**.

To use filter and select function you first to install and load dplyr
library, to install dplyr :

```
install.packages("dplyr")
```

** note: you need to run the previous command only once for your machine as this command will download then install the dplyr package and there will be no need to run this command each time you want to use the dplyr library.*

Then you need to load the dplyr library - as we did in the datasets library - using the following command:

```
library(dplyr)
```

Selection is selecting some attributes of interest, for example we might be interested of students grades and names only when recording their marks, while we would need to know their birth dates when matching sorting their ages.
We select columns using the **[ ]** operator.

```
Ages = dataframe[‘age’]
```

***Note**: age must be included in single or double quotes because it is a literal not a variable.*

We can select multiple columns, by passing them as a vector, list or array.

***Note**: here we use the combine function that is written as c*

```
grades = dataframe[c(‘name’,’final grade’)]
```

Or alternatively we can use the **select** function

```
grades = select(dataFrame,’name’,’final grade’)
```

Filtering is selecting some instances according to a certain criteria or condition.for example we might be interested of students with A grades only rather than all students, or to search for a student of a certain name.We select We can also use **filter** function

```
grades = filter(dataFrame,final grade ==’A’)
```

Indexing is selecting rows or columns according to their index
We index columns,rows using the **[ ]** operator.

```
firstStudent = dataframe[1,‘name’]

secondStudent = dataframe[2,‘name’]

top5 = dataframe[1:5,‘name’]

firstAndThird = dataframe[c(1,3),‘name’]
```

Sorting we can sort a dataframe based on a certain column using order function and indexing.

- Ascending order:

  ```
  students = dataframe[order(dataframe[‘grade’]),]
  ```

- Descending Order:

  ```
  students = dataframe[order(-dataframe[‘grade’]),]
  ```

Application

1. Load the iris dataset using the next command.

   ```
   library(datasets)
   data("iris")
   ```

   Now you can use `iris` as the name of the data variable

   ex: to display the iris data columns names: `names(iris)`

2. Display the head of the iris data frame.
3. Display only Sepal length,width and flower type.
4. Display the head of each type of flower separately.
5. Display Flowers whose sepal length is above average only.
6. Display The top 5 flowers according to petal width.

Hint: *to find Mean of a column in a dataframe use the `$` operator. The `$` allows you extract elements by name from a named list.*

ex*: `mean(iris$Petal.Length)` outputs* ***3.758***